inter-application data connection protocol known as remote procedure call (or RPC)(other well known protocols, such as asynchronous message queuing protocols, can also be used). Each publisher application could be running on a separate machine, alternatively, a single machine could be running a plurality of publisher applications. The broker network 2 is made up of a plurality of distribution agents (21 through 27) which are connected in a hierarchical fashion which will be described below as a "tree structure". These distribution agents, each of which could be running on a separate machine, are data processing applications which distribute data messages through the broker network 2 from publishers to subscribers. Subscriber applications 31, 32, 33 and 34 connect to the broker network 2 via RPC in order to receive published messages.

Please replace the paragraph 1, on page 4, with the following rewritten paragraph:

Publishers 110 and 120 first connect via RPC directly to a root distribution agent 21 which in turn connects via RPC to second level distribution agents 22 and 23 which in turn connect via RPC to third level distribution agents 24, 25, 26 and 27 (also known as "leaf distribution agents" since they are the final distribution agents in the tree structure). Each distribution agent could be running on its own machine, or alternatively, groups of distribution agents could be running on the same machine. The leaf distribution agents connect via RPC to subscriber applications 31 through 34, each of which could be running on its own machine.

Please replace the paragraph 1, on page 6, with the following
rewritten paragraph:

In the Fig. 1, a publisher application 110, running on one
computer, is, for example, a supplier of live stock market data
quotes. That is, publisher application 110 provides frequent
messages stating the present value of share prices. In this
example, publisher application 110 is publishing messages on a
stream called "stock" which has already been configured in the
broker network 2. As is well known, when publisher 110 wishes to
publish a stock quote message to stream "stock", publisher 110
makes an RPC call to the root distribution agent 21 which is at
the top level of the broker network tree structure. In this
example, subscriber application 32, running on another computer,
has sent a subscription request via an RPC call to leaf
distribution agent 24, which is at the bottom level of the tree
structure, indicating that subscriber 32 would like to subscribe
to stream "stock".

Please replace the paragraph 2, on page 6 and continuing
onto page 7, with the following rewritten paragraph:

Thus, whenever publisher 110 publishes a data message to
stream "stock" the distribution tree structure of broker network
2 channels the message down through the root distribution agent
21, through any intermediary distribution agents (e.g., 22 in the

example of Fig. 1) and through the leaf distribution agent 24 to the subscriber 32. This involves a series of RPC calls being made between each successive circle in the diagram of Fig. 1 connecting publisher 110 and subscriber 32 (i.e., 110 to 21, 21 to 22, 22 to 24 and 24 to 32).

Please replace paragraph 2, on page 10 and continuing onto page 11, with the following rewritten paragraph:

According to one aspect, the present invention provides a message broker data processing apparatus including: a unit for receiving published messages on a topic from a plurality of publisher applications; a unit for processing the received messages; and a unit for distributing the processed messages to a subscriber application; where the unit for receiving includes a plurality of publication point data processing nodes, each of which receives published messages on the topic from a publisher application.

Please replace the paragraph 2, on page 13 and continuing onto page 14, with the following rewritten paragraph:

In Fig. 3 a message broker 320 receives published messages on a topic called "IBM stock" from a publisher application 310a (which is an application running at a major stock exchange in the United States of America) and distributes such published messages

to subscriber application 33 (which is a stock broking agency also located in the United States of America) which has previously registered a subscription to the topic "IBM stock". Message broker 320 also receives published messages on the topic "IBM stock" from another publisher application 310b (which is an application running at a major stock exchange in the United Kingdom) and distributes such published messages to subscriber application 33 (again, which is a stock broking agency located in the United States of America) which has previously registered a subscription to the topic "IBM stock". In this example, the publisher application, broker and subscriber applications are all running on separate machines (and are thus interconnected via a network which is not shown in Fig. 3). In other embodiments, however, two or more of the applications (e.g., the publisher and the broker) could be running on the same machine. Further, as was explained above, the broker 320 is most likely running on a plurality of machines.

Please replace the paragraph 1, on page 14, with the following rewritten paragraph:

When one of the publisher applications 310a or 310b communicates with the broker 320 in order to publish messages thereto, the publisher application specifies a particular

publication point (e.g., 323 or 324) as the point of entry into the message broker 32. A publication point data processing node (or "publication point" for short) is a data processing node which acts as a point of entry for published messages in a messageflow of data processing nodes making up a message broker. That is, each publication point is at the beginning of a specific data processing path through the broker. A publisher application selects a publication point depending on which particular desired path the published messages should take depending on the nature of the published messages and the nature of the processing that will be carried out on that path.

Please replace the paragraph 2, on page 14 and continuing onto page 15, with the following rewritten paragraph:

For example, publisher application 310b selects publication point 324 because publisher application 310b is located in the United Kingdom and thus publisher application 310b "knows" that a message transformation will be needed. Specifically, once the publisher application 310b's published messages pass through the publication point 324, they are passed to message transformation data processing node 321 which performs the function of transforming the format of the published messages so that the IBM stock prices, which are originally published in UK pounds by publisher application 310b, are converted to US dollars. The message transformation node 321 accesses local storage 322 in order to determine the current exchange rate of UK pounds to US dollars (this exchange rate is updated at the beginning of every business day). After having their UK pound amounts converted to

US dollars, the messages are output from the message transformation node 321 and received at a subscription point processing node 325.

Please replace the paragraph 2, on page 16, with the following rewritten paragraph:

A9

On the other hand, publisher application 310a communicates with the broker 320 via another publication point 323, and thus published messages from publisher 310a take another path through the broker bypassing the message transformation data processing node 321. Specifically, the published messages from publisher 310a are sent directly to subscription point data processing node 325. Publisher application 310a chooses to communicate with publication point 323 because publisher application 310a is located in the United States and thus the published messages are already in the US dollars format, and thus there is no need to transform the messages to the US dollars format, which is the format required by the subscriber application 33. Subscription point data processing node 325 then performs a publish/subscribe topic matching operation and determines that subscriber application 33 has previously entered a subscription request to the topic "IBM stock". Thus, subscription point processing node 325 distributes the published messages to subscriber application 33.

Please replace the paragraph 1, on 17, with the following rewritten paragraph:

A 10

Thus, by the use of a plurality (two in Fig. 3) of publication point data processing nodes in a message broker, publisher applications can select amongst the plurality of publication points in order to publish messages which will be received by subscribers in a message format selected by the subscriber without having to use different topics (the topic "IBM stock" is the same for both publication points 323 and 324 and for both publisher applications 310a and 310b). This allows access control to be easily carried out on a topic basis. For example, the broker can perform a security measure on both publisher applications 310a and 310b by simply checking whether the requested topic "IBM stock" of their published messages is a topic which has previously been determined as acceptable for publishers 310a and 310b from a security standpoint.